22

Blockchain Applications and What's Next

Blockchain technology is evolving rapidly, and it will continue to affect the way we conduct our day-to-day business as new developments emerge. It has challenged existing business models and promised great benefits, such as cost-saving, efficiency, and transparency. So far in this book, we've explored the technical underpinnings of blockchain technology, such as cryptography, consensus mechanisms, and distributed systems concepts.

This chapter will explore some use cases and applications of blockchain technology in the **Internet of Things** (**IoT**) and the finance, government, and media sectors, and we will learn how blockchain technology can address challenges in these industries and others. Moreover, we'll learn about the latest developments, emerging trends, issues, and future predictions related to blockchain technology. We'll also present some topics related to open research problems and improvements to blockchain technology. Along the way, we'll cover the following main topics:

- Use cases, including IoT, government, health, and **Artificial Intelligence** (**AI**)
- Some emerging trends
- Some challenges

Use cases

Digital currencies were the first-ever application of blockchain technology, which arguably didn't realize the technology's full potential. Although **Bitcoin**, the first major blockchain conceptualization, was introduced in 2008, it was not until years later that blockchain technology's possible applications beyond cryptocurrencies were realized. In 2010, the discussion started regarding BitDNS, a decentralized naming system for domains on the internet.

Then, **Namecoin** (`https://en.bitcoinwiki.org/wiki/Namecoin`) was started in April 2011 with a different vision to Bitcoin, the sole purpose of which was to provision electronic cash. This can be considered the first example of blockchain usage other than purely as a cryptocurrency.

Next, in 2013, the first **Initial Coin Offering** (**ICO**), **MasterCoin**, emerged, which paved the way for more ICOs. After that, Ethereum's ICO was hugely successful. With the availability of Ethereum in 2015, a general-purpose smart contract platform, more ideas started to emerge around various applications of blockchain technology. Since then, many use cases of blockchain technology in various industries have been proposed. In this chapter, five main industries have been selected due to their popularity and promising blockchain use cases for discussion:

- IoT
- Government
- Health
- Media

Let's begin with one of the most exciting use cases of blockchain technology: IoT.

IoT

IoT has recently gained a lot of traction due to its potential for transforming business applications and everyday life. IoT can be defined as a network of computationally intelligent physical objects (any objects, such as cars, fridges, and industrial sensors) that are capable of connecting to the internet, sensing real-world events or environments, reacting to those events, collecting relevant data, and communicating this over the internet.

This simple definition has enormous implications and has led to exciting concepts, such as wearables (such as health trackers or watches), smart homes, smart grids, smart connected cars, and smart cities, which are all based on this basic concept of an IoT device. One thing to note is that all these components are accessible and controllable over the internet, via the IoT.

After dissecting the definition of IoT, four functions come to light as being performed by an IoT device: **sensing**, **reacting**, **collecting**, and **communicating**. All these functions are performed by using various components on the IoT device. Sensing is performed by **sensors**. Reacting or controlling is performed by **actuators**; collecting is the function of various sensors, and communication is performed by chips, which provide network connectivity.

In the next section, we'll introduce the typical architecture of an IoT-based ecosystem.

IoT architecture

A typical IoT architecture can consist of many physical objects connected to each other, and to a centralized cloud server. This is shown in the following diagram:

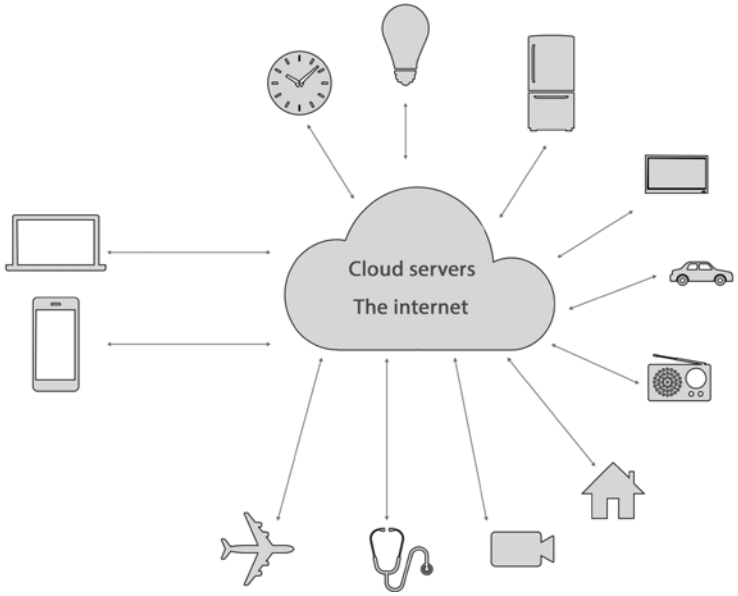


Figure 22.1: A typical IoT network

Elements of the IoT are spread across multiple layers, and various existing reference architectures can be used to develop IoT systems. A five-layer model can be used to describe IoT, which contains a **physical object** layer, a **device** layer, a **network** layer, a **services** layer, and an **application** layer. Each layer or level is responsible for various functions and includes multiple components. These are shown in the following figure:

Application layer Transportation, financial, insurance, and many others
Management layer Data processing, analytics, security management
Network layer LAN, WAN, PAN, routers
Device layer Sensors, actuators, smart devices
Physical objects People, cars, homes

Figure 22.2: IoT layered model

Now, let's examine each layer in detail.

The physical object layer

This layer includes any real-world physical objects. It includes objects like cars, fridges, trains, and homes. In fact, anything that is required to be monitored and controlled can be connected to the IoT.

The device layer

This layer contains things that make up the IoT, such as sensors, transducers, actuators, smartphones, smart devices, and **Radio-Frequency Identification** (**RFID**) tags. There can be many categories of sensors, such as body sensors, home sensors, and environmental sensors, based on the type of work they perform. This layer is the core of the IoT ecosystem where various sensors are used to sense real-world environments. This layer includes sensors that can monitor temperature, humidity, liquid flow, chemicals, air, pressure, and much more. Usually, an **Analog-to-Digital Converter** (**ADC**) is required for a device to turn the real-world analog signal into a digital signal that a microprocessor can understand.

Actuators in this layer provide the means to enable the control of external environments; for example, starting a motor or opening a door. These components also require digital-to-analog converters to convert a digital signal into analog. This method is especially relevant when control of a mechanical component is required by the IoT device.

The network layer

This layer is composed of various network devices that are used to provide internet connectivity between devices and the cloud, or servers that are part of the IoT ecosystem. These devices can include gateways, routers, hubs, and switches. This layer can include two types of communication.

First, there are the horizontal means of communication, which include radio, Bluetooth, Wi-Fi, Ethernet, LANs, Zigbee, and PANs, and can be used to provide communication between IoT devices. This layer can optionally be included in the device layer as it physically resides on the device layer where devices can communicate with each other.

Second, we have communication to the next layer, which is usually done through the internet and provides communication between machines and people or other upper layers.

The management layer

This layer provides the management layer for the IoT ecosystem. This includes platforms that process data gathered from IoT devices and turning it into meaningful insights. Device management, security management, and data flow management are included in this layer. It also manages communication between the device and application layers.

The application layer

This layer includes the applications running on top of the IoT network. This layer can consist of many applications, depending on the requirements, such as transportation, healthcare, finance, insurance, or supply chain management tasks. This list, of course, is not exhaustive by any stretch of the imagination; there is a myriad of IoT applications that can fall into this layer.

With the availability of cheap sensors, hardware, and bandwidth, IoT has gained popularity in recent years and currently has applications in many different areas, including healthcare, insurance, supply chain management, home automation, industrial automation, and infrastructure management. Moreover, advancements in technology such as the availability of IPv6, smaller and more powerful processors, and better internet access have also played a vital role in the popularity of IoT.

IPv6 is the latest version of the internet protocol, which, with a size of 128 bits, offers more address space compared to the 32-bit IPv4.

In the next section, we'll discuss some advantages of IoT and blockchain convergence.

Benefits of IoT and blockchain convergence

The benefits of IoT with blockchain technology range from saving costs to enabling businesses to make vital decisions, and thus improve performance based on the data provided by the IoT devices. Even in domestic usage, IoT-equipped home appliances can provide valuable data for cost savings. For example, smart meters for energy monitoring can provide valuable information on how energy is being used and can convey that back to the service provider. Raw data from millions of IoT devices is analyzed and provides meaningful insights that help in making timely and efficient business decisions.

The usual IoT model is based on a centralized paradigm, where IoT devices usually connect to a cloud infrastructure or central servers to report and process the relevant data. This centralization is somewhat vulnerable to exploitation, including hacking and data theft. Moreover, not having control of personal data on a single, centralized service provider also increases the possibility of security and privacy issues. While there are methods and techniques for building a highly secure IoT ecosystem based on the normal IoT model, there are much more specific and desirable benefits that a blockchain-based model can bring to the IoT.

Blockchain technology for IoT can help to build trust, reduce costs, and accelerate transactions. Additionally, decentralization, which is at the very core of blockchain technology, can eliminate single points of failure in an IoT network. For example, a central server may be unable to cope with the amount of data that billions of IoT devices (things) produce at high frequency, whereas a decentralized blockchain serving as a communication layer between these IoT devices enables all the IoT devices in the blockchain to maintain a copy of the data store of their own, which means that this architecture is inherently highly available and resilient.

Some IoT devices losing their database does not result in the entire network coming to a standstill, as might be the case with a centralized server, even with a disaster recovery solution. Also, the **Peer-to-Peer** (**P2P**) communication model provided by blockchain technology can help reduce costs because there is no need to build high-cost centralized data centers or implement complex public key infrastructure for security. Devices can communicate with each other directly or via routers.

As an estimate made by various researchers and companies, by 2025, there will be roughly 41 billion devices connected to the internet. With this explosion of billions of devices connecting to the internet, it is hard to imagine that centralized infrastructures will be able to cope with the high demands of bandwidth, services, and availability without incurring excessive expenditure. A blockchain-based IoT will be able to solve scalability, availability, privacy, and reliability issues in the current IoT model.

Blockchain technology enables things to communicate and transact with each other directly, and with the availability of smart contracts, negotiation and financial transactions can also occur directly between devices instead of requiring an intermediary, an authority, or human intervention. For example, if a room in a hotel is vacant, it can rent itself out, negotiate the rent, and can open the door lock for a human who has paid the required fee. Another example could be that if a washing machine runs out of detergent, it could order it online after finding the best price and value based on the logic programmed into its smart contract.

The aforementioned five-layer IoT model can be adapted to a blockchain-based model by adding a blockchain layer on top of the network layer. This layer will run smart contracts and provide security, privacy, integrity, autonomy, scalability, and decentralization services to the IoT ecosystem. The management layer, in this case, will just consist of software related to analytics and processing, and security and control can be moved to the blockchain layer.

This new model with six layers is visualized in the following table:

Application layer Transportation, finance, insurance, and many other domains
Management layer Data processing, analytics, security management
Blockchain layer Security, consensus, P2P (M2M) autonomous transactions, decentralization, smart contracts
Network layer LAN, WAN, PAN, routers
Device layer Sensors, actuators, smart devices
Physical objects People, cars, homes

Figure 22.3: Layered blockchain-based IoT model

In this model, the other layers are expected to remain the same, but an additional blockchain layer will be introduced as middleware between all participants of the IoT network.

It can also be visualized as a P2P IoT network after abstracting away all the layers mentioned earlier. This model is shown in the following diagram, where all the devices are communicating and negotiating with each other without a central command and control entity:

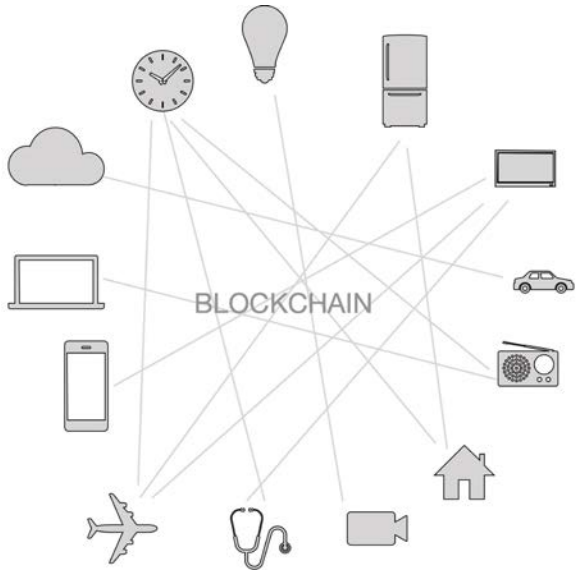


Figure 22.4: Blockchain-based direct (P2P or machine-to-machine (M2M)) communication model

An IoT-based blockchain can also result in cost savings due to the easier device management offered by a blockchain-based, decentralized approach. The IoT network can also be optimized for performance by using blockchain technology. In this case, there will be no need to store IoT data centrally for millions of devices. This is because storage and processing requirements can be distributed to all IoT devices on the blockchain. This can result in completely removing the need for large data centers for processing and storing IoT data.

A blockchain-based IoT can also thwart **denial-of-service** (**DoS**) attacks: hackers can target a centralized server or data center more efficiently, but with blockchain's distributed and decentralized nature, such attacks are no longer possible. Additionally, if, as estimated, there will soon be billions of devices connected to the internet, it will become almost impossible to manage security and updates for all those devices from traditional, centrally owned servers. Blockchain technology can provide a solution to this problem by allowing devices to communicate with each other directly in a secure manner, and even request firmware and security updates from each other. On a blockchain network, these communications can be recorded immutably and securely, which will provide auditability, integrity, and transparency to the system. This mechanism is not possible in traditional P2P systems.

In summary, there are clear benefits that can be reaped due to the convergence of IoT and blockchain, and a lot of research and work in academia and across industries are already in progress. Practical use cases and platforms have emerged in the form of **Platform as a Service** (**PaaS**) for blockchain-based IoT, such as the IBM Watson IoT blockchain. There are various projects that have already been proposed that provide blockchain-based IoT solutions, such as IBM Blue Horizon and IBM Bluemix, two examples of IoT platforms that support blockchain IoT. Various start-ups, such as **Filament**, have already proposed novel ideas about how to build a decentralized network that enables devices on the IoT to transact with each other directly and autonomously, driven by smart contracts. In the next section, we discuss the implementation of a blockchain-based IoT system.

Implementing blockchain-based IoT in practice

In the following section, a practical example is provided of how to build a simple IoT device and connect it to the Ethereum blockchain. The IoT device in this example will be connected to the Ethereum blockchain and used to open a door (in this case, the door lock is represented by a **Light Emitting Diode** (**LED**) when the appropriate amount of funds is sent by a user. Note that this is a simple example and requires a more rigorously tested version to be implemented in production.

The prerequisite hardware components are listed as follows:

- A **Raspberry Pi** device, which is a **Single Board Computer** (**SBC**). The Raspberry Pi is an SBC developed as a low-cost computer to promote computer education, but it has also gained much more popularity as the tool of choice for building IoT platforms. You may be able to use earlier models too, but those have not been tested. You can explore more about Raspberry Pi at `https://www.raspberrypi.com/products/`.
- **LED**: An LED can be used as a visual indication for an event.
- **Resistor**: A 330-ohm component is required, which provides resistance to the passing current based on its rating. It is not necessary to understand the theory behind this for this experiment; any standard electronics engineering text covers all these topics in detail.
- **Breadboard**: This provides a means of building an electronic circuit without requiring soldering.
- **T-shaped cobbler**: This is inserted on the breadboard, as shown in the following figure, and provides a labeled view of all **General-Purpose I/O** (**GPIO**) pins for the Raspberry Pi.
- **Ribbon cable connector**: This is simply used to provide connectivity between the Raspberry Pi and the breadboard via GPIO.

All these components are shown in the following figure:

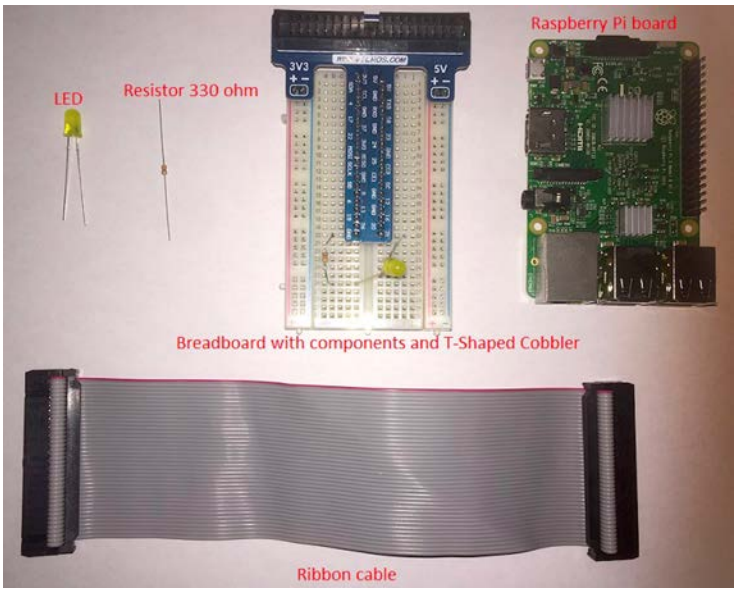


Figure 22.5: Required components

Setting up Raspberry Pi

First, the Raspberry Pi needs to be set up. This can be done by using **New Out Of Box Software** (**NOOBS**), which provides an easy method of installing Raspbian or any other OS.

NOOBS is a user-friendly and easy-to-use OS installation manager for the Raspberry Pi.

NOOBS can be downloaded and installed from `https://www.raspberrypi.org/downloads/noobs/`.

Alternatively, you can just install Raspbian from `https://www.raspberrypi.org/downloads/raspbian/`.

Another alternative, available at `https://github.com/debian-pi/raspbian-ua-netinst`, can also be used to install a minimal non-GUI version of the Raspbian OS.

For this example, NOOBS has been used to install Raspbian. As such, the rest of the exercise assumes Raspbian is installed on the SD memory card of the Raspberry Pi. The command output in the following screenshot shows which architecture the OS is running on.

In this case, it is `armv71`; therefore, the ARM-compatible binary for Geth will be downloaded.

The platform can be confirmed by running the command `uname -a` in a terminal window in the Raspberry Pi Raspbian OS:

```
$ uname -a
Linux raspberrypi 4.4.34-v7+ #930 SMP Wed Nov 23 15:20:41 GMT 2016 armv71 GNU/
Linux
```

Once the Raspbian OS has been installed, the next step is to download the appropriate Geth binary for the Raspberry Pi ARM platform.

The download and installation steps are described in detail as follows:

1. First, download the Geth binary on Raspberry Pi. We use `wget` to download the `geth` client images:

```
$ wget https://gethstore.blob.core.windows.net/builds/geth-linux- arm7-
1.5.6-2a609af5.tar.gz
```

Note that, in this example, a specific version of Geth is being downloaded. Other versions are available, which can be downloaded from `https://geth.ethereum.org/downloads/`. However, it's recommended that you download the version that has been used in the examples in this chapter.

2. Unzip and extract this into a directory. The directory named `geth-linux-arm7-1.5.6-2a609af5` will be created automatically with the `tar` command, as shown here:

```
$ tar -zxvf geth-linux-arm7-1.5.6-2a609af5.tar
```

3. This command will create a directory named `geth-linux-arm7-1.5.6-2a609af5` and will extract the Geth binary and related files into that directory. The Geth binary can be copied into `/usr/bin` or the appropriate path on Raspbian to make it available from anywhere in the OS. When the download is finished, the next step is to create the genesis block.
4. The same genesis block should be used that we created previously in *Chapter 10*, *Ethereum in Practice*. The genesis file can be copied from the other node on the network. This is shown in the following code segment:

```
{
     "nonce": "0x0000000000000042",
     "timestamp": "0x00",
     "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
     "extraData": "0x00",
     "gasLimit": "0x8000000",
     "difficulty": "0x0400",
          "mixhash": "0x0000000000000000000000000000000000000000000000000000000000000000",
     "coinbase": "0x3333333333333333333333333333333333333333",
     "alloc": {
     },
     "config": {
         "chainId": 786,
         "homesteadBlock": 0,
         "eip155Block": 0,
         "eip158Block": 0
     }
}
```

Alternatively, an entirely new genesis block can be generated. Elements of the genesis file were explained in the *Private nets* section of *Chapter 9*, *Ethereum Architecture*, which you can refer to as a refresher.

5. Once the `genesis.json` file has been copied onto the Raspberry Pi, the following command can be run to generate the genesis block. It is important that the same genesis block is used that was generated previously; otherwise, the nodes will effectively be running on separate networks:

```
$ ./geth init genesis.json
```

6. This will create the genesis block and initialize the chain.

7. After genesis block creation, we add peers to the network. This can be achieved by creating a list of nodes on our private network. In order to define the list, we create a file named `static-nodes.json` using any text editor and add the `enode` of the peer that `geth`, on the Raspberry Pi, will connect to for synchronization.
8. The `enode` URL can be obtained from the Geth JavaScript console by running the following command, which should be run on the peer to which Raspberry Pi is going to connect:

```
> admin.nodeInfo
```

9. This will show an output similar to the one shown in the following screenshot:

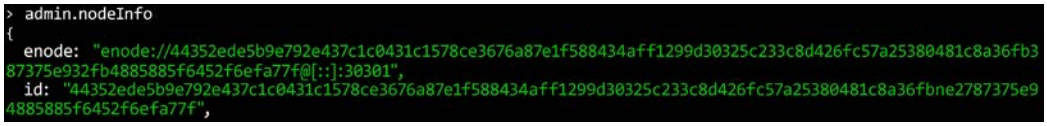


Figure 22.6: Geth nodeInfo

10. The `static-nodes.json` file should contain the `enode` value, as shown in the following screenshot. We can view the contents using the `cat` command:

```
$ cat static-nodes.json
```

11. This command will produce the output shown here:

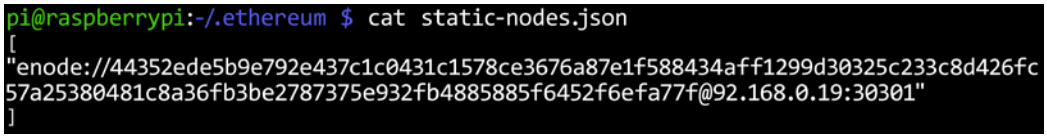


Figure 22.7: Static node configuration

After this step, further instructions presented in the following sections can be followed to connect Raspberry Pi to the other node on the private network. In this example, the Raspberry Pi will be connected to the network with the ID 786 that we created in *Chapter 10*, *Ethereum in Practice*. The key is to use the same genesis file created previously and different port numbers. Using the same genesis file will ensure that clients connect to the same network in which the genesis file originated. Different ports are not a strict requirement; however, if the two nodes are running in a private network and access from an environment external to the network is required, then a combination of a DMZ, a router, and port forwarding will be used. Therefore, it is recommended to use different TCP ports to allow port forwarding to work correctly.

Setting up the first node

First, the `geth` client needs to be started on the first node using the following command. The `--identity` switch allows an identifying name to be specified for the node:

```
$ geth --datadir .ethereum/privatenet/ --networkid 786 --maxpeers 5 --http
--http.api web3,eth,debug,personal,net --http.port 9001 --http.corsdomain "*"
--port 30301 --identity "drequinox"
```

This will start up the `geth` node. Once the node has been started up, it should be kept running, and another `geth` instance should be started from the Raspberry Pi node.

Setting up the Raspberry Pi node

On Raspberry Pi, the following command is required to be run to start `geth` and to sync it with other nodes (in this case, only one node). The following is the command:

```
$ ./geth --networkid 786 --maxpeers 5 --http --http.api
web3,eth,debug,personal,net --http.port 9002 --http.corsdomain "*" --port 30302
--identity "raspberry"
```

This should start up the `geth` node on the Raspberry Pi. Note that when the output log contains a row displaying `Block synchronisation started`, this means that the node has connected successfully to its peer that we started up earlier.

This connectivity can be further verified by running commands in the `geth` console on both nodes, as shown in the following screenshot. The `geth` client can be attached by simply running the following command on the Raspberry Pi:

```
$ geth attach
```

This will open the JavaScript `geth` console for interacting with the `geth` node. We can use the `admin.peers` command to see the connected peers:

```
> admin.peers
[{
    caps: ["eth/62", "eth/63"],
    id: "44352ede5b9e792e437c1c0431c1578ce3676a87e1f588434aff1299d30325c233c8d426fc57a25380481c8a36fb3be2787375e932f
b4885885f6452f6efa77f",
    name: "Geth/drequtnox/v1.5.2-stable-c8695209/ltnux/go1.7.3",
    network: {
      localAddress: "192.168.0.21:56550",
      remoteAddress: "192.168.0.19:30301"
    },
    protocols: {
      eth: {
        difficulty: 117194155397,
        head: "0x2d32c90b4c9dacea9a109b0ae52c1ebf511915bb618a2d3c55a80a63852e89f6",
        version: 63
      }
    }
}]
```

Figure 22.8: Geth console admin peers command running on the Raspberry Pi

Similarly, we can attach to the `geth` instance by running the following command on the first node:

```
$ geth attach ipc:.ethereum/privatenet/geth.ipc
```

Once the console is available, `admin.peers` can be run to reveal the details about other connected nodes, as shown in the following screenshot:

Figure 22.9: Geth console admin peers command running on the other peer

Once both nodes are up and running, further prerequisites can be installed to set up the experiment. The installation of Node.js and the relevant JavaScript libraries is required.

Installing Node.js

The required libraries and dependencies are listed here. First, Node.js and `npm` need to be updated on the Raspberry Pi Raspbian OS. For this, the following steps can be followed:

1. Install Node.js on the Raspberry Pi using the following command:

```
$ curl -sL https://deb.nodesource.com/setup_7.x | sudo -E bash -
```

Note that we are using Node version 7.x for this example, simply for demonstration. You can use a later version if desired.

This should install Node.js.

2. Run the update via `apt-get`:

```
$ sudo apt-get install nodejs
```

3. Verification can be performed by running the following command to ensure that the correct versions of Node.js and `npm` have been installed, as shown in the following screenshot:

```
$ npm -v
4.0.5

$ node -v
v7.4.0
```

It should be noted that these versions are not a necessity; any of the latest versions of `npm` and Node.js should work. However, the examples in this chapter make use of `npm 4.0.5` and `node v7.4.0` simply for demonstration, so it is recommended that you use the same versions to avoid any compatibility issues.

4. Install Ethereum web3 using the following command:

```
$ npm install web3@0.18.0
```

5. This should install Web3. Web3 is required to enable JavaScript code to access the Ethereum blockchain.

Make sure that the specific version of Web3 shown in the snippet, or a version similar to this (for example, 0.20.2), is installed instead of the default, version 1.2.11 (at the time of writing). As this example was originally developed and tested using version 0.18.0, it is recommended that a Web3 0.20.2 or 0.18.0 stable version should be used for this example. The aim of this example, which is to show how Ethereum-based IoT networks can be created, doesn't change regardless of whether a newer or older version of Web3 is used. You can install the recommended version by using:

```
$ npm install web3@0.20.2
```

6. Similarly, `onoff` can be installed, which is required to communicate with the Raspberry Pi and control GPIO:

```
$ npm install onoff --save
```

7. This will install `onoff`.

In this section, we set up a private network with our Raspberry Pi, and another node with `geth`, `nodejs`, web3, and the `onoff` library to run our example on.

When all the prerequisites have been installed, the hardware setup can be performed. For this purpose, a simple circuit will be built using a breadboard and a few electronic components.

Building the electronic circuit

As shown in the following figure, the **positive** leg (long leg) of the **LED** is connected to pin number **21** of the **breadboard**, and the **negative** (short leg) is connected to the **resistor**, which is then connected to the **ground** (**GND**) pin of the **breadboard**. Once the connections have been set up, the **ribbon cable** can be used to connect to the GPIO connector on the Raspberry Pi:

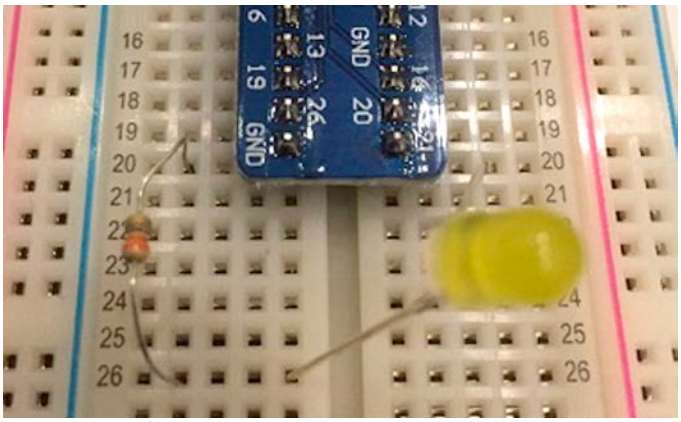


Figure 22.10: Connections for components on the breadboard

Once the connections have been set up correctly and the Raspberry Pi has been updated with the appropriate libraries and Geth, the next step is to develop a simple smart contract that expects a value. If the value provided is not what it expects, it does not trigger an event; otherwise, if the value passed matches the correct value, the event triggers, which can be read by the client JavaScript program running via Node.js.

Developing and running a Solidity contract

Of course, the **Solidity** contract can be very complicated and can also deal with the **ether** sent to it. If the amount of ether is equal to the required amount, then the event can trigger. However, in this example, the aim is to demonstrate the usage of smart contracts to trigger events, which can then be read by JavaScript running on Node.js and can then, in turn, trigger actions on IoT devices using various libraries.

The smart contract source code is shown as follows:

```
pragma solidity ^0.4.0;
contract simpleIOT {
    uint roomrent = 10;
    event roomRented(bool returnValue);
    function getRent (uint8 x) public returns (bool) {
        if (x == roomrent) {
            emit roomRented(true);
            return true;
        }
    }
}
```

The online Solidity compiler (the **Remix IDE**) can be used to run and test this contract. The **Application Binary Interface** (**ABI**) required for interacting with the contract is also available in the Remix IDE's Solidity compiler section.

The following is the ABI of the contract:

```
[
    {
        "constant": false,
        "inputs": [
            {
                "name": "x",
                "type": "uint8"
            }
        ],
        "name": "getRent",
        "outputs": [
            {
                "name": "",
                "type": "bool"
            }
        ],
        "payable": false,
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "anonymous": false,
        "inputs": [
            {
                "indexed": false,
                "name": "returnValue",
                "type": "bool"
            }
        ],
        "name": "roomRented",
        "type": "event"
    }
]
```

There are two methods by which the Raspberry Pi node can connect to the private blockchain via the web3 interface. The first is where the Raspberry Pi device is running its own `geth` client locally and maintains its ledger, as shown in the following diagram:

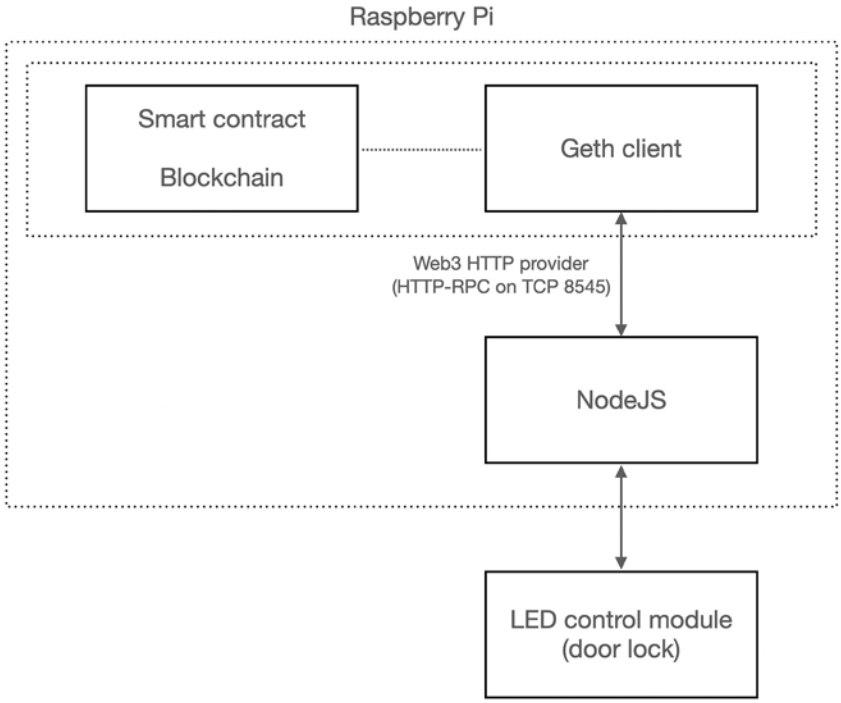


Figure 26.11: Application architecture of the room rental IoT application (an IoT device with a local ledger)

Sometimes, with resource-constrained devices, it is not possible to run a full `geth` node or even a light node. In that case, the second method, which uses a web3 provider (via an HTTP-RPC server), can be used to connect to the appropriate RPC channel. The following diagram shows the high-level architecture of an IoT application where the Raspberry Pi does not run a Geth instance. Instead, it runs only the minimum software required to provide IoT application functionality to the IoT device.

For blockchain-relevant operations, it connects to another blockchain node running on the network via HTTP-RPC:

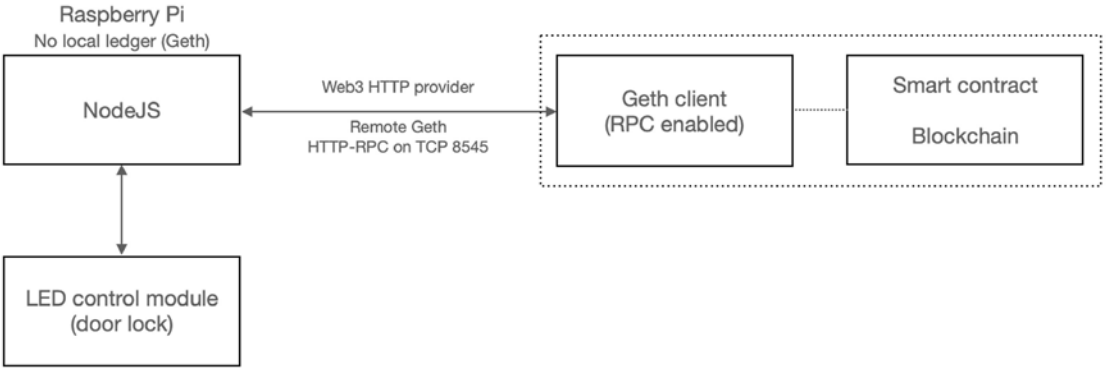


Figure 22.12: Application architecture of the room rental IoT application (an IoT device without a local ledger)

There are obvious security concerns that arise from exposing RPC interfaces publicly; therefore, it is recommended that this option is used only on private networks. If required to be used on public networks, appropriate security measures need to be put in place, such as allowing only the known IP addresses to connect to the `geth` RPC interface. This can be achieved by a combination of disabling peer discovery mechanisms and HTTP-RPC server listening interfaces.

More information about this can be found using `geth help`. Traditional network security measures such as **firewalls**, **Transport Layer Security** (**TLS**), and **certificates** can also be used but have not been discussed in this example for brevity. Now, **Truffle** can be used to deploy the contract on the private network ID `786` to which, at this point, the Raspberry Pi is connected.

Truffle `deploy` can be performed simply by using the following command; it is assumed that `truffle init` and other preliminaries discussed in *Chapter 12*, *Web3 Development Using Ethereum*, have already been performed:

```
$ truffle migrate
```

This should deploy the contract.

Once the contract has been deployed correctly, JavaScript code can be developed that will connect to the blockchain via web3, listen for the events from the smart contract in the blockchain, and turn the LED on via the Raspberry Pi. The JavaScript code required for this example is shown here. Simply write or copy and paste the code shown here into a file named `index.js`:

```
var Web3 = require('web3');
if (typeof web3 !== 'undefined')
{
    web3 = new Web3(web3.currentProvider);
}else
{
```

```
    web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:9002"));
    //http-rpc-port
}
var Gpio = require('onoff').Gpio;
var led = new Gpio(21,'out');
var coinbase = web3.eth.coinbase;
var ABIString =
'[{"constant":false,"inputs":[{"name":"x","type":"uint8"}],"name":"getRent",
"outputs":[{"name":"","type":"bool"}],
"payable":false,"stateMutability":"nonpayable",
"type":"function"},{"anonymous":false,
"inputs":[{"indexed":false,"name":"returnValue",
"type":"bool"}],"name":"roomRented","type":"event"}]';
var ABI = JSON.parse(ABIString);
var ContractAddress = '0x975881c44fbef4573fef33cccec1777a8f76669c';
web3.eth.defaultAccount = web3.eth.accounts[0];
var simpleiot = web3.eth.contract(ABI).at(ContractAddress);
var event = simpleiot.roomRented( {}, function(error, result) { if (!error)
{
    console.log("LED On");
    led.writeSync(1);
}
});
```

Note that, in the preceding code example, the contract address `0x975881c44fbef4573fef33cccec1777a8f76669c` for the `var ContractAddress` variable is specific to the deployment; it will be different when you run this example. Simply change the address in the file to what you see after deploying the contract.

Also, note the HTTP-RPC server listening port on which Geth has been started on Raspberry Pi. By default, it is TCP port `8545`. Remember to change this according to your Raspberry Pi setup and Geth configuration. It is set to `9002` in the preceding example code because Geth running on Raspberry Pi is listening on `9002` in the example. If it's listening on a different port on your Raspberry Pi, then change it to that port:

```
web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:9002"));
```

When Geth starts up, it shows which port it has the HTTP endpoint listening on. This is also configurable with `--http.port` in `geth` by specifying the port number value as a parameter for the flag.

This JavaScript code can be placed in a file on the Raspberry Pi; for example, `index.js`. It can be run by using the following command:

```
$ node index.js
```

This will start the program, which will run on Node.js and listen for events from the smart contract. Once the program is running correctly, the smart contract can be invoked by using the Truffle console. In this case, the `getRent` function is called with the parameter `10`, which is the expected value:

```
[truffle(development)> simpleiot.getRent(10)
'0x71f550949a4c5168af7b9f7f84fada99bccc20a123779642e5e8c0c0127266ee'
```

After the contract has been mined, `roomRented` will be triggered, which will turn on the LED.

In this example, it is a simple LED, but it can be any physical device, such as a room lock, that can be controlled via an actuator. If all works well, the LED will be turned on as a result of the smart contract's function invocation, as shown in the following figure:

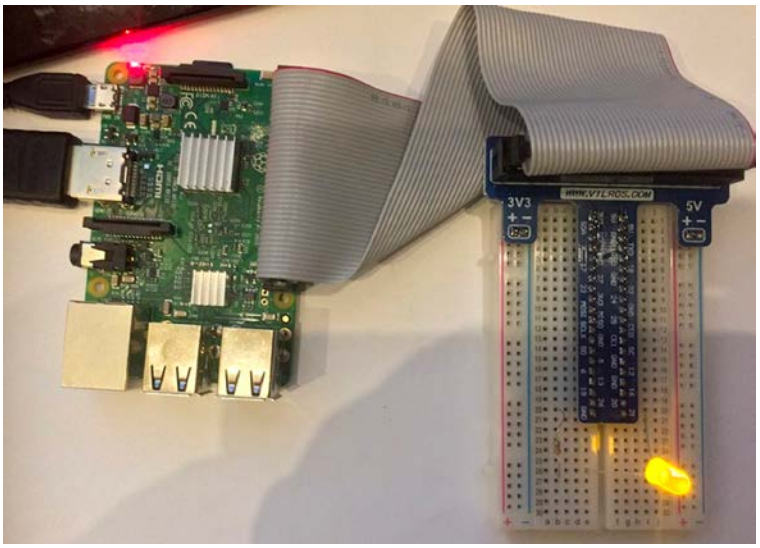


Figure 22.13: Raspberry Pi with LED control

Also, on the node side, it will display an output similar to the one shown here:

```
$ node index.js
LED On
```

As demonstrated in the preceding example, a private network of IoT devices can be built that runs a `geth` client on each of the nodes, listens for events from smart contracts, and triggers an action accordingly. The example shown is simple in design but demonstrates the underlying principles of an Ethereum network that can be built using IoT devices, along with smart contract-driven control of the physical devices.

In the next few sections, other applications of the blockchain will be discussed, such as in government, finance, media, and health.

Government

There are various applications of blockchain technology being researched currently that can support government functions and take the current model of e-government to the next level. First, in this section, some background for e-government will be provided, and then a few use cases, such as e-voting, homeland security (border control), and electronic IDs (citizen ID cards), will be discussed.

E-government, or electronic government, is a paradigm where IT and communication technology are used to deliver public services to citizens. The concept is not new and has been implemented in various countries around the world, but with blockchain technology, a new avenue of exploration has opened up. Many governments are researching the possibility of using blockchain technology for managing and delivering public services, including, but not limited to, identity cards, driving licenses, secure data sharing among various government departments, and contract management. Transparency, auditability, and integrity are attributes of blockchain technology that can go a long way in effectively managing various government functions. One of the government functions is border control, which we cover next.

Border control

Automated border control systems have been in use for decades now to thwart illegal entry into countries and prevent terrorism and human trafficking. Machine-readable travel documents, specifically **biometric passports**, have paved the way for automated border control; however, current systems are limited to a certain extent and blockchain technology can provide solutions. A **Machine-Readable Travel Document** (**MRTD**) standard is defined in document 9303 (`https://www.icao.int/publications/pages/publication.aspx?docnum=9303`) of the **International Civil Aviation Organization** (**ICAO**) and has been implemented by many countries around the world.

Each passport contains various security and identity attributes that can be used to identify the owner of the passport and circumvent attempts at tampering with these passports. These include biometric features such as retina scans, fingerprints, facial recognition, and standard ICAO-specified features, including a **Machine-Readable Zone** (**MRZ**) and other text attributes that are visible on the first page of the passport.

One key issue with current border control systems is data sharing, whereby the systems are controlled by a single entity, and data is not readily shared among law enforcement agencies. This inability to share data makes it challenging to track suspicious travel documents or individuals. Another issue is related to the immediate implementation of blacklisting a travel document; for example, when there is an immediate need to track and control suspicious travel documents. Currently, there is no mechanism available to blacklist or revoke a suspicious passport immediately and broadcast it to border control ports worldwide.

Blockchain technology can provide a solution to this problem by maintaining a blacklist in a smart contract that can be updated as required. Any changes will be immediately visible to all agencies and border control points, thus enabling immediate control over the movement of a suspected travel document.

It could be argued that traditional mechanisms like **Public Key Infrastructures** (**PKIs**) and P2P networks can also be used for this purpose, but they do not provide the benefits that a blockchain can provide. With a blockchain, the whole system can be simplified without the requirement of complex networks and PKI setups, which will also result in cost reduction. Moreover, blockchain-based systems will provide cryptographically guaranteed immutability, which helps with auditing and discourages any fraudulent activity.

The full database of all travel documents may not be stored on the blockchain currently due to inherent storage limitations, but a backend distributed database such as **BigchainDB**, **IPFS**, or **Swarm** can be used for that purpose. In this case, a hash of the travel document with the biometric ID of an individual can be stored in a simple smart contract, and a hash of the document can then be used to refer to the detailed data available on the distributed filesystem, such as IPFS. This way, when a travel document is blacklisted anywhere on the network, that information will be available immediately with the cryptographic guarantee of its authenticity and integrity throughout the distributed ledger. This functionality can also provide adequate support in anti-terrorism activities, thus playing a vital role in the homeland security function of a government.

A simple contract in **Solidity** can have an array defined for storing identities and associated biometric records. This array can be used to store the identifying information about a passport. The identity can be a hash of the MRZ of the passport or travel document concatenated with the biometric record from the RFID chip. A simple Boolean field can be used to identify blacklisted passports. Once this initial check passes, further detailed biometric verification can be performed by traditional systems. Eventually, when a decision is made regarding the entry of the passport holder, that decision can be propagated back to the blockchain, thus enabling all participants on the network to immediately share the outcome of the decision.

A high-level approach to building a blockchain-based border control system can be visualized as shown in the following diagram. In this scenario, the passport is presented for scanning to an RFID and page scanner, which reads the data page and extracts machine-readable information, along with a hash of the biometric data stored in the RFID chip. At this stage, a live photo and retina scan of the passport holder is also taken. This information is then passed on to the blockchain, where a smart contract is responsible for verifying the legitimacy of the travel document by first checking its list of blacklisted passports, and then requesting more data from the backend IPFS database for comparison. Note that the biometric data, such as a photo or retina scan, is not stored on the blockchain; instead, only a reference to this data in the backend (IPFS or BigchainDB) is stored in the blockchain.

If the data from the presented passport matches with what is held in the IPFS as files or in BigchainDB and passes the smart contract logical check, then the border gate can be opened:

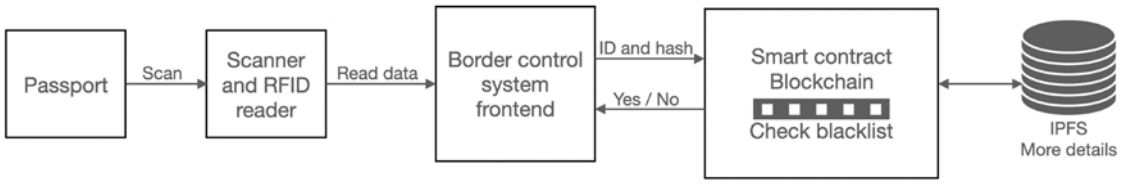


Figure 22.14: Automated border control using a blockchain

After verification, this information is propagated throughout the blockchain and is instantly available to all participants on the border control blockchain. These participants can be a worldwide consortium of homeland security departments of various nations. Another important government function is elections, which we cover next.

Elections

Voting in any government is a key function and allows citizens to participate in the democratic election process. While voting has evolved into a much more mature and secure process, it still has limitations that need to be addressed to achieve the desired level of maturity. Usually, the limitations in current voting systems revolve around fraud, weaknesses in operational processes, and especially transparency. Over the years, secure voting mechanisms (machines) have been built to ensure security and privacy, but they still have vulnerabilities that could be exploited. These vulnerabilities can lead to serious implications for the whole voting process and can result in mistrust of the government by the public.

Blockchain-based voting systems can resolve these issues by introducing end-to-end security and transparency into the process. Security is provided in the form of the guaranteed integrity and authenticity of votes by using public key cryptography, which comes as standard in a blockchain. Moreover, the immutability guaranteed by a blockchain ensures that votes cast once cannot be cast again. This can be achieved through a combination of biometric features and a smart contract maintaining a list of votes already cast. For example, a smart contract can maintain a list of already-cast votes with biometric IDs (for example, a fingerprint) and can use that to detect and prevent double-casting. Secondly, **Zero-Knowledge Proofs** (**ZKPs**) can also be used on the blockchain to protect voters' privacy. With a ZKP, a voter can remain anonymous by hiding their identity, and the vote itself can be kept confidential.

Recently, presidential elections were held in Sierra Leone using blockchain technology, making it the first country to use blockchain technology for elections: `https://www.coindesk.com/sierra-leone-secretly-holds-first-blockchain-powered-presidential-vote/`.

Another example is voting in the city of Zug. A report on this can be found here: `https://www.stadtzug.ch/_docn/1938568/eVoting_Final_Report_ENG.pdf`.

Citizen identification

Electronic IDs or national ID cards are issued by various countries around the world at present. These cards are secure and possess many security features that thwart duplication or tampering attempts. However, with the advent of blockchain technology, several improvements can be made to this process.

Digital identity is not only limited to just government-issued ID cards; it is a concept that applies to online social networks and forums, too. There can be multiple identities being used for different purposes. A blockchain-based online digital identity allows control over personal information sharing. Users can see who has used their data and for what purpose, as well as control access to it. This is not possible with the current infrastructure, which is centrally controlled.

The key benefit is that a single identity issued by the government can be used easily and transparently for multiple services via a single government blockchain. In this case, the blockchain serves as a platform where a government is providing various services, such as pensions, taxation, or benefits, and a single ID is being used to access all these services. The blockchain, in this case, provides a permanent record of every change and transaction made by a digital ID, thus ensuring the integrity and transparency of the system. Also, citizens can notarize birth certificates, marriages, deeds, and many other documents on the blockchain tied with their digital ID as proof of existence. Proof of existence provides digital evidence that something or some data exists or existed in the past at a specific date and time.

Currently, there are successful implementations of identity schemes in various countries that work well, and there is an argument that perhaps the blockchain is not required for identity management systems. Although there are several benefits, such as privacy and the control of the use of identity information, due to the current immaturity of blockchain technology, perhaps it is not ready for use in real-world identity systems. However, research is being carried out by various governments to explore the use of blockchains for identity management. We'll cover more on identity in *Chapter 20*, *Decentralized Identity*.

Moreover, laws such as the right to be forgotten can be quite difficult to incorporate into the blockchain due to its immutable nature.

Other government functions where blockchain technology can be implemented to improve cost and efficiency include the collection of taxes, benefits management and disbursement, land ownership record management, life event registration (marriages, births), motor vehicle registration, and licenses. This is not an exhaustive list, and, over time, many functions and processes of a government could be adapted to a blockchain-based model. The key benefits of blockchain, such as immutability, transparency, and decentralization, can help to bring improvements to most of the traditional government systems.

Now, let's see how the health industry can benefit from blockchain technology.

Health

The health industry has also been identified as another major industry that can benefit from adopting blockchain technology. The blockchain can provide an immutable, auditable, and transparent system that traditional P2P networks cannot. Also, the blockchain provides a simpler, more cost-effective infrastructure compared to traditional complex PKI networks. In healthcare, major issues such as privacy compromises, data breaches, high costs, and fraud can arise from a lack of interoperability, overly complex processes, transparency, auditability, and control. Another burning issue is counterfeit medicine; especially in developing countries, this is a major cause of concern.

With the adaptability of the blockchain in the health sector, several benefits can be realized, including cost savings, increased trust, faster processing of claims, high availability, no operational errors due to complexity in the operational procedures, and the prevention of the distribution of counterfeit medicine.

From another angle, blockchains that provide a digital currency as an incentive for mining can be used to provide processing power to solve scientific problems. This helps to find cures for certain diseases. Examples include **FoldingCoin**, which rewards its miners with FLDC tokens for sharing their computer's processing power for solving scientific problems that require unusually large calculations.

FoldingCoin is available at `http://foldingcoin.net/`.

Another similar project is called **CureCoin,** which is available at `https://www.curecoin.net/`. It is yet to be seen how successful these projects will be in achieving their goals, but the idea is very promising.

Next, we'll discuss media, another use case where blockchain can provide a cheap, fair, and transparent media ecosystem.

Media

Critical issues in the media industry revolve around content distribution, rights management, and royalty payments to artists. For example, digital music can be copied many times without any restrictions, and any attempts to apply copy protection have been hacked in some way or another. There is no control over the distribution of the content that a musician or songwriter produces; it can be copied as many times as needed without any restriction, and consequently has an impact on royalty payments. Also, payments are not always guaranteed and are based on traditional airtime figures. All these issues revolving around copy protection and royalty payments can be resolved by connecting consumers, artists, and all players in the industry, allowing transparency and control over the process.

The blockchain can provide a network where digital music is cryptographically guaranteed to be owned only by the consumers who pay for it. This payment mechanism is controlled by a smart contract instead of a centralized media agency or authority. The payments will be automatically made based on the logic embedded within the smart contract and the number of downloads.

A recent example of such an initiative is Musicoin: `https://musicoin.org`.

Another example of a decentralized NFT marketplace is `https://www.nftchain.tech`.

Moreover, illegally copying digital music files can be stopped altogether because everything is recorded and owned immutably in a transparent manner on the blockchain. A music file, for example, can be stored with owner information and a timestamp that can be traced throughout the blockchain network. Furthermore, consumers who own a legal copy of some content are cryptographically tied to the content they have, and it cannot be moved to another owner unless permitted by the owner. Copyrights and transfers can be managed easily via a blockchain once all digital content is immutably recorded on that blockchain. Smart contracts can then control the distribution and payment to all concerned parties.

Blockchain and AI

AI is a field of computer science that endeavors to build intelligent agents that can make rational decisions based on the scenarios and environment that they observe around them. Machine learning plays a vital role in AI technology by making use of raw data as a learning resource. A key requirement in AI-based systems is the availability of authentic data that can be used for machine learning and model building. Therefore, the explosion of data coming out of IoT devices, smartphones, and other means of data acquisition means that AI and machine learning is becoming more and more powerful. There is, however, a requirement for data authenticity, which is where the convergence with the blockchain comes in. Once consumers, producers, and other entities are on a blockchain, the data that is generated as a result of interaction between these entities can be readily used as input for machine learning engines with a guarantee of authenticity.

The convergence of the blockchain with IoT was discussed earlier in this chapter. Briefly, it can be said that due to the authenticity, integrity, privacy, and shared architecture of blockchain technology, IoT networks would benefit greatly from making use of it. This can be realized in the form of an IoT network that runs on a blockchain and makes use of a decentralized **mesh network** for communication in order to facilitate M2M communication in real time.

A mesh network is a network topology that allows all nodes on a network to connect with one another in a cooperative and dynamic fashion, to facilitate the efficient routing of data.

All of the data that is generated as a result of M2M communication can be used in **machine learning** processes to augment the functionality of artificially intelligent DAOs or simple **Autonomous Agents** (**AAs**). These AAs can act as agents in a blockchain-provided **Distributed Artificial Intelligence** (**DAI**) or distributed machine learning environment, which can learn over time using **machine learning** processes. This would enable them to make better decisions for the good of the blockchain.

Moreover, blockchain technology can play a vital role in federated learning (a sub-field of machine learning) by enhancing security, privacy, trust, and decentralization. In federated learning, data is distributed across multiple devices and used to train a shared model, without having to share raw data. By using a blockchain, the data-sharing process can be made secure and accessible only to authorized parties. Additionally, the blockchain can incentivize data providers to contribute data and computing power to train the model by rewarding them with tokens. The blockchain also ensures the transparency and immutability of the training data and model parameters, making it tamper-proof and auditable by multiple parties. The decentralized management of the model can also be achieved using a blockchain, ensuring that all participants have access to the latest version of the model and are working toward the same goal.

It could also be argued that if an IoT device were hacked, it would send malformed data to the blockchain. This issue would be mitigated using blockchain technology because an IoT device would be part of the blockchain (as a node) and would have the same security properties applied to it as a standard node in the blockchain network.

These properties include the incentivization of good behavior, the rejection of malformed transactions, the strict verification of transactions, and various other checks that are part of blockchain protocol. Therefore, even if an IoT device is hacked, it would be treated as a **Byzantine node** by the blockchain network and would not cause any adverse impact on the network. The possibility of combining intelligent oracles, intelligent smart contracts, and AAs will give rise to **Artificially Intelligent Decentralized Autonomous Organizations** (**AIDAOs**) that can act on behalf of humans to run entire organizations on their own. This is another side of AI that could potentially become normal in the future. However, more research is required to realize this vision.

The convergence of blockchain technology with various other fields, such as 3D printing, virtual reality, augmented reality, spatial computing, and the gaming industry, has also been envisaged. For example, in a multiplayer online game, a blockchain's decentralized approach allows more transparency and can ensure that no central authority is gaining an unfair advantage by manipulating game rules. Each of these topics is currently an active area of research, and more interest and development are expected. Blockchain technology is going to change the world. The revolution has already begun!

There are many applications of blockchain technology and, as discussed in this section, they can be implemented in various industries to bring about solutions to existing problems.

In the next section, we will discuss some trends that are emerging with the evolution of blockchain technology.

Some emerging trends

With the advancement of blockchain technology and the rigorous research being conducted in this space, various developments have been seen in recent years, including application-specific blockchains, new start-ups, and standardization efforts to improve adoption. Some other trends include research on interoperability between chains, scalability, privacy, and security research. Limitations around the **interoperability** of blockchains have also resulted in changes oriented toward the development of systems that can work across multiple blockchains. Some examples of these systems are Interledger, Polkadot, Cosmos, cross-chain atomic swaps, relay networks, and blockchain bridges. This trend of addressing technical challenges is expected to continue to develop in years to come, and even if almost all fundamental challenges are addressed in blockchain technology, further enhancements and optimization research will continue.

Blockchain technology has stimulated intense research interest, both in academia and the commercial sector. In recent years, interest has dramatically increased, and now major institutions and researchers around the world are exploring this technology. This growth in interest is primarily because blockchain technology can help to make businesses and their operations more efficient, cheaper to run, and more transparent. Technologies such as **ZKPs**, fully **homomorphic encryption**, and functional encryption are being researched for their potential use in blockchains. Already, ZKPs have been implemented for the first time at a practical level in the form of Zcash, which has addressed privacy issues in cryptocurrency networks. It's evident that blockchains and cryptocurrencies have helped with the advancement of cryptography, especially financial cryptography.

New fields of research are emerging with blockchains, most notably cryptoeconomics, which is the study of protocols governing the decentralized digital economy. With the continuing development of blockchains and cryptocurrencies, research in this area has also grown.

When it was realized in 2010 that existing methods were no longer efficient for mining bitcoins, miners shifted toward optimizing the available mining hardware. These initial efforts included the use of GPUs, and then **Field-Programmable Gate Arrays** (**FPGAs**) were used after GPUs reached their hash rate limit. Very quickly after that, **Application-Specific Integrated Circuits** (**ASICs**) emerged, which increased mining power significantly. This trend is expected to grow further with research into optimizing ASICs by parallelizing and decreasing their die sizes. Such optimizations will make ASICs even faster and more efficient. Moreover, GPU programming initiatives are also expected to grow, with the regular emergence of new cryptocurrencies, many of which make use of PoW algorithms that can benefit from GPU processing capabilities. For example, recently Zcash has spurred interest in GPU mining rigs and related programming using NVIDIA CUDA and OpenCL. The aim of Zcash is to use multiple GPUs in parallel to optimize mining operations. Also, some research has been done in the field of trusted computing hardware, such as Intel's **Software Guard Extensions** (**SGX**) to address security issues on blockchains. Intel's SGX has also been used in a novel consensus algorithm called **Proof of Elapsed Time** (**PoET**), which was discussed in *Chapter 14*, *Hyperledger*. Hardware research and development are expected to continue, and soon many more hardware applications will be explored. Some examples include **physical unclonable functions**, the acceleration of blockchain processes (such as cryptography), IoT hardware, and hardware security module integration for key management. Moreover, as the prover times in zero-knowledge systems are not ideal, there has been some research on building **ASICs**, FPGAs, and GPUs for improving prover times in systems like SNARKs and so on.

With the realization of security issues and vulnerabilities in smart contract programming languages, and generally in blockchain protocols, there is now a keen interest in the formal verification and testing of smart contracts and other blockchain components before production deployments. For this, various efforts are already underway, many of which we discussed in *Chapter 19*, *Blockchain Security*.

There is also an increased interest in the development of programming languages for developing smart contracts. Efforts are more focused on domain-specific languages, for example, **Solidity** for Ethereum and **Pact** for Kadena. This is just a start, and many new languages are likely to be developed as the available technology advances. Languages for writing circuits like Circom, Zokrates, Cairo, and noir are now also available. This trend will continue.

With the current maturity level of cloud platforms, many companies have started to provide **Blockchain as a Service** (**BaaS**). The most prominent examples are Microsoft's Azure, Google Cloud, Oracle, in which the blockchain platform is provided as a service, and IBM Cloud, which provides IBM's blockchain as a service. This trend is only expected to grow in the next few years, and more companies will likely emerge that provide BaaS to consumers. The convergence of other technologies with the blockchain offers major benefits. At their core, blockchains provide resilience, security, and transparency, which, when combined with other technologies, results in a very powerful complementary technology. For example, IoT stands to gain major benefits such as integrity, decentralization, and scalability when implemented via a blockchain.

AI is expected to gain numerous benefits from the implementation of blockchain technology: in fact, within blockchain technology, AI can be implemented in the form of AAs, which can make rational decisions on behalf of humans. Some ideas of the benefits that the convergence of AI and blockchain can provide are listed as follows:

- The blockchain can share machine learning models in a secure and **P2P** manner.
- The blockchain can serve as an auditing layer for decisions made by AI.
- As AI and machine learning capabilities grow, the blockchain can provide a mechanism to monitor and control any malicious behavior that AI may manifest. While AI can be used for good, it can also be used by malicious actors. In this case, the blockchain can serve as a control mechanism to thwart any attacks. For example, all agents on a blockchain can be controlled under a consensus mechanism, and malicious Byzantine behavior can be handled as per the blockchain consensus protocol.
- AI can also benefit blockchain technology by enabling artificially intelligent smart contracts.
- AI can be applied to other components of a blockchain to achieve, for example, adaptive consensus mechanisms. These can, based on network conditions, readjust fault-tolerance requirements and as a result enable a more efficient consensus mechanism.

There are of course many more examples of innovative applications of the blockchain in combination with the aforementioned technologies. Some of these were discussed in detail in the *Blockchain and AI* section of this chapter.

Some challenges

Apart from security, scalability, and privacy, which were discussed at length in earlier chapters, several other obstacles should be addressed before the broader adoption of blockchains can be achieved. We'll introduce some of these more specific areas to address in the following sections.

Regulation is considered one of the most significant challenges to blockchain development. The core issue is that blockchains, and their associated cryptocurrencies, are not recognized as legal systems or forms of currency by any government. Even though in some cases, blockchain tokens have been classified as having monetary value, for example in the US and Germany, they are still far from being accepted as regular currencies. Moreover, blockchains in their current state are not recognized by financial regulatory bodies as platforms that are stable or reliable enough to be used by financial institutions.

There have been, however, various initiatives proposed by regulatory authorities around the world to research and set regulations. Bitcoin in its current state is fully unregulated, even though some attempts have been made by governments to tax it. In the UK, under the EU VAT directive, Bitcoin transactions are exempt from **Value-Added Tax** (**VAT**). This may change after the finalization of Brexit, but **Capital Gains Tax** (**CGT**) may still be applicable in some scenarios. Some attempt by financial regulatory authorities to regulate blockchain technology is expected very soon, especially after the recent announcement by the UK's **Financial Conduct Authority** (**FCA**) that it may approve some companies that are using blockchain technology for their business services.

Another regulatory concern is that blockchain technology is not ready for production deployments. Even though the Bitcoin blockchain has evolved into a solid blockchain platform and is used in production, it is not suitable for every scenario. This is especially true in the case of sensitive environments such as finance and health. However, this situation is changing rapidly, and this chapter has already explored various examples of new blockchain projects that have been implemented in real life, such as the ASX blockchain post-trade solution. This trend is expected to grow as efforts are made to improve the associated technology and address technical limitations such as scalability and privacy.

Security is also another general concern that has been highlighted by many researchers, which is especially applicable to the finance and health sectors. A report by the **European Union Agency for Network and Information Security** (**ENISA**) has highlighted some distributed ledger-specific concerns, including the need for regulation, auditing, control, and governance. The report is available at `https://www.enisa.europa.eu/news/enisa-news/enisa-report-on-blockchain-technology-and-security`.

Some concerns highlighted in the report also include smart contract management, key management, **Anti-Money Laundering** (**AML**) regulations, and anti-fraud tools. Clearly, while blockchain is generally seen as a solution to many technical business challenges, it can also be misused. In the next section, we'll see some scenarios in which cryptocurrencies and blockchain networks are being used for illegal activities.

With the key attributes of censorship resistance and decentralization, blockchain technology can help to improve transparency and efficiency in many walks of life. However, the somewhat unregulated nature of blockchain technology means that it has the potential to be used for illegal activity. For example, consider a scenario where illegal content is published over the internet. Normally, it can be immediately shut down by approaching the relevant authorities and website service providers, but this is not possible in blockchains. Once something is there on the blockchain, it is almost impossible to remove it. This means that any unacceptable content, once published on the blockchain, cannot be removed. If the blockchain is used for distributing immoral or illegal content, then there is no way for anyone to prevent it. This fact poses a serious challenge, and it seems that some regulation and control would be beneficial in this scenario, but it provokes the critical question: how can a blockchain be regulated? In this case, it may not be prudent to create regulatory laws first and then see if blockchain technology adapts, because this might disrupt innovation and progress. It may be more sensible to let blockchain technology grow first in a similar, organic manner to the internet. Then, when its user base reaches a critical mass, governing bodies can call for the application of regulations around the implementation and usage of blockchain technology.

There are various real-life examples where the **dark web** is used in conjunction with **Bitcoin** to perform illegal activities.

The dark web is a term used to describe different networks that exist on the internet but necessitate the use of special hardware, programs, configuration, or credentials for access. More on the dark web can be found here: `https://en.wikipedia.org/wiki/Dark_web`.

One example is the Silk Road online marketplace, which was used to sell illegal drugs and other contraband over the internet. It used Bitcoin for payments and was hosted on the dark web using URLs only visible with a browser called Tor. Although the Silk Road was shut down after years of effort by law enforcement agencies, other sites have emerged that offer similar services; as such, this activity remains a major concern. Imagine that an illegal website was hosted on IPFS, a P2P distributed and decentralized storage network built on a blockchain; there would be no easy way of shutting it down. It is clear that the absence of control and regulation can encourage illegal activity, meaning criminal enterprises like Silk Road will keep appearing. Further development of totally anonymous transaction capabilities such as Zcash, while useful in various legitimate scenarios, could provide another layer of protection for criminals. Clearly, it depends on who is using the technology; anonymity can be good in many scenarios, for example, in the health industry, where patient records should be kept private and anonymous. However, it may not be appropriate if it can also be used by criminals to hide their activities. One solution might be to introduce intelligent bots, AAs, or even contracts that are programmed and embedded with regulatory logic. They would most likely be programmed by regulators and law enforcement agencies and live on the blockchain as a means to provide governance and control. For example, a blockchain could be designed in such a way that every smart contract has to go through a controller contract, which scrutinizes the code logic and provides a regulatory mechanism to control the behavior of the contract.

It may also be possible to get each smart contract's code to be inspected by regulatory authorities, and once a smart contract's code has a certain level of authenticity attached to it in the form of certificates issued by a regulator, it will be deployed on the blockchain network. This concept of **binary signing** is akin to the already established concept of **code signing**, whereby executables are digitally signed as a means to confirm that the code is bona fide and not malicious. This idea is more applicable in the context of semi-private or regulated blockchains, where a certain degree of control is required by a regulatory authority, for example, in finance. It means, however, that some degree of trust is required to be placed in a trusted third-party regulator, which may not be desirable due to the deviation from full decentralization. However, to address this, the blockchain itself can be used to provide a decentralized, transparent, and secure certificate issuing and digital signing mechanism.

There also needs to be a fine balance between privacy and transparency. Too much transparency can result in undermining personal privacy; however, too much privacy can result in the loss of transparency. The choice between privacy and transparency depends on the use case; however, a state of equilibrium is highly desirable to provide the best solution to handle both requirements.

It is envisaged that other technologies, such as IoT and AI, will converge for the mutual benefit and wider adoption of both the blockchain and the other given technology.

Summary

In this chapter, five main industries that can benefit from blockchain technology were discussed. First, IoT was discussed, which is another revolutionary technology in its own right. By combining it with blockchain technology, several limitations can be addressed, which would bring about tremendous benefits to the IoT industry. More focus has been given to IoT in this chapter, as it is the most prominent and most ready candidate for adapting blockchain technology.

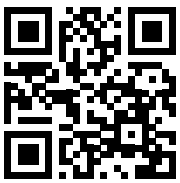
Moreover, applications in the government sector were discussed, whereby various government processes such as homeland security, identification cards, and benefits disbursements can be made transparent, secure, and more robust. Furthermore, issues in the finance sector were discussed, along with possible solutions that blockchain technology could provide. Finally, some aspects of the health sector and music industry were also discussed. All these use cases and many more stand on the pillars provided by the core attributes of blockchain technology, such as decentralization, transparency, reliability, and security. After this, a few trends were discussed that are expected to continue as the technology progresses.

In this book, we have studied the technical foundations of the blockchain and learned how to build practical real-world decentralized applications. We've explored cryptocurrencies, alternative blockchains, enterprise blockchains, scalability, security, privacy, identity, and DeFi, and examined the underlying mechanics of blockchain technology. We've combined these theoretical foundations with the practical development, design, and deployment of blockchain networks, smart contracts, and decentralized applications. We learned how to use Truffle, Ganache, Drizzle, and other tools and techniques to build DApps. Furthermore, detailed accounts of tokenization, consensus mechanisms, and recent developments such as Post-Merge Ethereum were provided.

The blockchain can arguably be considered the most innovative technology of this decade, and as we've seen throughout this book, it is one of the most active areas of study by researchers, academics... and now, you! You are now capable of applying the knowledge from this book and continuing your learning to try your hand at entry-level blockchain development or architecture. Given the material provided in this book, you are now equipped with the necessary skills and knowledge of blockchain technology to become an entry-level blockchain engineer and participate in further research and development concerning this amazing technology and be part of the blockchain revolution!

Join us on Discord!

To join the Discord community for this book – where you can share feedback, ask questions to the author, and learn about new releases – follow the QR code below:



`https://packt.link/ips2H`